IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): David R. Wardwell

Appln. No.: 10/529,701

Filed: March 29, 2005

Title: Method and system for collating data in a distributed computer network

Art Unit: 2452

Examiner: D. Chankong

Attorney Docket No.: 96764

Customer No.: 22242

Confirmation No. 2061

This Appeal Brief is being electronically filed using the USPTO's EFS-Web on November 3, 2010

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

In response to the Final Rejection mailed May 13, 2010 and Notice of Appeal filed September 8, 2010 and pursuant to 37 C.F.R. §41.37, we hereby respectfully submit the following Brief in support of the corresponding appeal.

Table of Contents

(1) Real Party in Interest

The real party in interest is Wisterium Development LLC, the assignee of record.

(2) Related Appeals and Interferences

There are no other related appeals, interferences, or judicial proceedings known to appellant, the appellant's legal representative, or assignee that will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

Claims 1-12 are pending and presently stand at least twice and finally rejected and constitute the subject matter of this appeal.

(4) Status of Amendments

No post-final amendments have been entered.

(5) Summary of Claimed Subject Matter

A concise explanation of the subject matter of the independent claims appears as follows (with corresponding references to the page and line number (in page:line-line or page:line-page:line format) of the specification as filed (and/or by paragraph number of the filed and published specification, where appropriate)) and to the drawing(s) (if any) by figure number and reference characters.[1]

Independent Claim Subject Matter Map

Independent Claim 1	
A method for collating data in a distributed computer network having non-synchronous compute nodes, said method comprising:	This preamble is considered to be non-limiting.
receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets are provided by individual ones of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets;	Figure 1; Page 3, lines 3-6 (published ¶7); Page 5, lines 5-18 (¶¶14-15); Page 11, lines 25-30 (¶28).
inserting said data packets into a software container according to predetermined rules for determining a logical order for said data packets;	Page 3, lines 6-7 (published ¶7); Page 6, lines 3-25 (¶¶17-20); Page 10, line 15 to page 13, line 17 (¶¶26-31).

[1] It will be understood that in some instances the content of a given referenced paragraph may additionally contain content that is tangential or even irrelevant to the claimed subject matter. It will also be understood that this summarization of the claimed subject matter is, in fact, a "summary" and that the applicant does not represent or intend that this brief presentation, or the accompanying references to the drawings and the specification, comprises an exhaustive presentation in this regard. As always, the claims are to be viewed and interpreted in view of the context of the entire specification sans the Abstract.

Independent Claim 1	
locating common groups of said data packets within said software container according to said predetermined rules;	Page 3, lines 7-9 (published ¶7); Page 6, lines 3-25 (¶¶17-20); Page 10, line 15 to page 13, line 17 (¶¶26-31).
protecting said software container against incomplete groups of said data packets according to a grouping criteria; and	Page 3, lines 9-11 (published ¶7); Page 6, lines 3-16 and page 7, lines 1-5 (¶¶17-18 and 22); Page 17, line 19 to page 18, line 22 (¶¶46-48).
outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met.	Page 3, lines 11-13 (published ¶7); Page 6, lines 3-16 and 26-28 (¶¶17-18 and 21); Page 13, line 19 to page 14, line 12 (¶¶32-35).

Independent Claim 5	
An apparatus for collating data in a distributed computer network having non-synchronous compute nodes, said apparatus comprising:	This preamble is considered to be non-limiting.

Independent Claim 5	
means for receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets are provided by individual ones of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets;	This is a means-plus-function element per 35 U.S.C. 112, sixth paragraph. Support for this recitation can be found at least at Figure 1, element 114; Page 3, lines 3-6 (published ¶7); Page 5, lines 5-18 (¶¶14-15); and Page 11, lines 25-30 (¶28). Page 18, lines 24-25 (¶49) further describes the subject matter as operating in a fully functional computer system. The software is described as a "MEQ" as supported at least at Page 6, line 3 through page 7, line 5 (¶¶17-22). The "MEQ" structure is described at Page 14, line 13 to page 17, line 17. Code for the software can be found at Page 8, line 6 to page 9, line 52; Page 12, line 10 to line 12 (¶¶36-45).
means for inserting said data packets into a software container according to predetermined rules for determining a logical order for said data packets;	This is a means-plus-function element per 35 U.S.C. 112, sixth paragraph. Support for this recitation can be found at least at Page 3, lines 6-7 (published ¶7); Page 6, lines 3-25 (¶¶17-20); Page 10, line 15 to page 13, line 17 (¶¶26-31). Page 18, lines 24-25 (¶49) further describes the subject matter as operating in a fully functional computer system. The software is described as a "MEQ" as supported at least at Page 6, line 3 to page 7, line 5 (¶¶17-22). Code for the software can be found at Page 8, line 6 to page 9, line 52; Page 12, line 10 to line 12.

Independent Claim 5	
means for locating common groups of said data packets within said software container according to said predetermined rules;	This is a means-plus-function element per 35 U.S.C. 112, sixth paragraph. Support for this recitation can be found at least at Page 3, lines 7-9 (published ¶7); Page 6, lines 3-25 (¶¶17-20); Page 10, line 15 to page 13, line 17 (¶¶26-31). Page 18, lines 24-25 (¶49) further describes the subject matter as operating in a fully functional computer system. The software is described as a "MEQ" as supported at least at Page 6, line 3 through page 7, line 5 (¶¶17-22). Code for the software can be found at Page 8, line 6 to page 9, line 52; Page 12, line 10 to line 12.
means for protecting said software container against incomplete groups of said data packets according to a grouping criteria; and	This is a means-plus-function element per 35 U.S.C. 112, sixth paragraph. Support for this recitation can be found at least at Page 3, lines 9-11 (published ¶7); Page 6, lines 3-16 and page 7, lines 1-5 (¶¶17-18 and 22); Page 17, line 19 to page 18, line 22 (¶¶46-48). Page 18, lines 24-25 (¶49) further describes the subject matter as operating in a fully functional computer system. The software is described as a "MEQ" as supported at least at Page 6, line 3 through page 7, line 5 (¶¶17-22). Code for the software can be found at Page 8, line 6 to page 9, line 52; Page 12, line 10 to line 12.

Independent Claim 5	
means for outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met.	This is a means-plus-function element per 35 U.S.C. 112, sixth paragraph. Support for this recitation can be found at least at Page 3, lines 11-13 (published ¶7); Page 6, lines 3-16 and 26-28 (¶¶17-18 and 21); Page 13, line 19 to page 14, line 12 (¶¶32-35). Page 18, lines 24-25 (¶49) further describes the subject matter as operating in a fully functional computer system. The software is described as a "MEQ" as supported at least at Page 6, line 3 through page 7, line 5 (¶¶17-22). Code for the software can be found at Page 8, line 6 to page 9, line 52; Page 12, line 10 to line 12.

Independent Claim 9	
A recordable type medium having a computer program product for collating data in a distributed computer network having non-synchronous compute nodes, said recordable type medium comprising:	This preamble is considered to be non-limiting. Support for a recordable type medium is at page 18, line 24 to page 19, line 2 (published ¶49).
computer program code for receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets are provided by individual ones of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets;	Figure 1; Page 3, lines 3-6 (published ¶7); Page 5, lines 5-18 (¶¶14-15); Page 11, lines 25-30 (¶28).

Independent Claim 9	
computer program code for inserting said data packets into a software container according to user predetermined rules for determining a logical order for said data packets;	Page 3, lines 6-7 (published ¶7); Page 6, lines 3-25 (¶¶17-20); Page 10, line 15 to page 13, line 17 (¶¶26-31).
computer program code for locating common groups of said data packets within said software container according to said user predetermined rules;	Page 3, lines 7-9 (published ¶7); Page 6, lines 3-25 (¶¶17-20); Page 10, line 15 to page 13, line 17 (¶¶26-31).
computer program code for protecting said software container against incomplete groups of said data packets due to system anomalies or quality of service within said distributed computer network according to a grouping criteria; and	Page 3, lines 9-11 (published ¶7); Page 6, lines 3-16 and page 7, lines 1-5 (¶¶17-18 and 22); Page 17, line 19 to page 18, line 22 (¶¶46-48).
computer program code for outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met.	Page 3, lines 11-13 (published ¶7); Page 6, lines 3-16 and 26-28 (¶¶17-18 and 21); Page 13, line 19 to page 14, line 12 (¶¶32-35).

(6) Grounds of Rejection to be Reviewed on Appeal

Whether claims 1-12 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 6,957,281 to Mann et al. ("Mann") in combination with U.S. Patent No. 6,907,041 to Turner et al. ("Turner").

(7) Argument

U.S. Patent No. 6,957,281 to Mann et al. ("Mann") in combination with U.S. Patent No. 6,907,041 to Turner et al. ("Turner").

Claims 1-12

Claims 1-12 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 6,957,281 to Mann et al. ("Mann") in combination with U.S. Patent No. 6,907,041 to Turner et al. ("Turner").

Mann teaches an arrangement "for ingress processing optimization via traffic classification and grouping [where a] plurality of packets are classified according to a classification criterion." Mann at Abstract. Mann describes the context of its teachings in its Background section as follows: "Data exchange between independent network nodes is frequently accomplished via establishing a 'session' to synchronize data transfer between the independent network nodes." Mann at col. 1, lines 19-22. The Examiner notes that "based on the teaching that the host system [of Mann] receives packets from different sessions, it is obvious that the host system receives packets from a plurality of network nodes (each a different session) in the network." Examiner's Answer of July 30, 2009 at page 7 (emphasis added). Mann then teaches that "the order of the received packets may be re-arranged in the packet queue 220 (e.g., arrange all the packets with a same session number in a sequence). . . . When classification is complete, all packets that are classified as a single group have, for example, the same session number . . . This group of packets may be delivered to the host 140 as one unit identified by the session number." Mann at col. 5, lines 22-25 and 36-41.

In contrast, claim 1 recites in combination with other subject matter "receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets is provided by one of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets" and "outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met." Because the system of Mann delivers groups of packets into one unit identified by the session number and each session number corresponds to one node, Mann cannot teach every element of claim 1.

The Examiner improperly proposes modifying Mann with the teachings of Turner to remedy this shortcoming. Because Mann's teachings are limited to that of specific node-to-node communication as discussed above, modifying Mann to include this very feature as suggested by the Examiner results in an improper modification of a principle of operation of Mann. See MPEP 2143.01 Section VI "The Proposed Modification Cannot Change the Principle of Operation of a Reference."

More specifically, Mann teaches in its background section that it is designed to address the issue of "[w]hen a plurality of network nodes simultaneously access a common network resource, packets from a communication session may be shuffled with packets from hundreds of other different sessions. Due to this unpredictable data shuffling, a host system generally processes each received packet individually, including identifying a session from the received packet and accordingly identifying a corresponding session on the host system to which the received packet belongs. There is an overhead on the host system associated with such process. . . . Furthermore, the overhead may increase drastically when there are a plurality of concurrent communication sessions. High overhead degrades a hosts system's performance." Mann at col. 1, lines 45-63. Mann is addressing the problems behind matching up one to one sessions. In other words, one of skill in the art would not start with Mann to formulate the subject matter of the pending claims when Mann would have to be modified in a fundamental way to combine data from multiple sessions. Such a

The Examiner argues that Mann may be modified because:

> *Mann* is directed towards grouping using a classification criterion to classify packets and grouping the packets having the same criterion into a packet bundle [abstract]. One example of this criterion is a packet's session number [column 3 «lines 44-46»: *"for example*, the classification-based packet transferring mechanism may classify 'received packets according to their session numbers I column 4 «lines 53-59»: "To classify received packets according to, *for example*, session numbers, the classification-based packet transferring mechanism 120... "].
>
> Based on the foregoing sections, it would have been clear to one of ordinary skill in the art that *Mann* was not limiting his classifications to merely session numbers. Thus, one of ordinary skill in the art would have been able to apply other methods of classifying packets in order to form a grouped packet bundle.

Office action of May 13, 2010 at page 3 (emphasis in original). The Examiner's logic here fails because modification of Mann to group packets together from more than one session, the modification proposed by the Examiner, runs counter to the problem being solved by Mann: "identifying a session from the received packet and accordingly identifying a corresponding session on the host system to which the received packet belongs." Mann at col. 1, lines 50-53. In this way, Mann teaches away from the modification proposed by the Examiner.

Because the modification of Mann proposed by the Examiner is taught away from by Mann and would result in a fundamental modification to the teachings of Mann, the obviousness rejection of claim 1 over Mann in view of Turner proposed by the Examiner should be reversed.

Independent claims 5 and 9 include limitations similar to that of claim 1. For all these reasons, we submit that claims 1, 5, and 9 are patentable over Mann. The remaining claims ultimately depend upon one of the independent claims shown allowable over Mann above. While we believe that other arguments are available to highlight the allowable subject matter presented in various ones of these dependent claims, we also believe that the comments set forth herein are sufficiently compelling to warrant exclusion of such additional points for the sake of brevity and expedited consideration.

For at least these reasons, we respectfully request that the rejections of claims 1-12 be reversed.

Conclusion

Claims 1-12 are in suitable condition to support allowance. We therefore respectfully seek a reversal of the Examiner's rejection of these claims.

Respectfully submitted,

FITCH, EVEN, TABIN & FLANNERY
Customer No. 22242

Dated: November 3, 2010
120 South LaSalle, Suite 1600
Chicago, Illinois 60603-3406
Telephone: (312) 577-7000

By /Nicholas T. Peters/
Nicholas T. Peters
Registration No. 53,456
ntpete@fitcheven.com

(8) Claims Appendix

1. A method for collating data in a distributed computer network having non-synchronous compute nodes, said method comprising:

receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets are provided by individual ones of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets;

inserting said data packets into a software container according to predetermined rules for determining a logical order for said data packets;

locating common groups of said data packets within said software container according to said predetermined rules;

protecting said software container against incomplete groups of said data packets according to a grouping criteria; and

outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met.

2. The method of Claim 1, wherein said inserting further includes inserting said data packets into said software container according to individual packet time reference.

3. The method of Claim 1, wherein said locating further includes locating common groups of said data packets within said software container according to individual packet time reference.

4. The method of Claim 1, wherein said outputting further includes outputting said data packets in respective logical groups that represent time-synchronous packets from said non-synchronous compute nodes after said grouping criteria has been met.

5. An apparatus for collating data in a distributed computer network having non-synchronous compute nodes, said apparatus comprising:

means for receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets are provided by individual ones of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets;

means for inserting said data packets into a software container according to predetermined rules for determining a logical order for said data packets;

means for locating common groups of said data packets within said software container according to said predetermined rules;

means for protecting said software container against incomplete groups of said data packets according to a grouping criteria; and

means for outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met.

6. The apparatus of Claim 5, wherein said means for inserting further includes means for inserting said data packets into a software container according to individual packet time reference.

7. The apparatus of Claim 4, wherein said means for locating further includes means for locating common groups of said data packets within said container according to individual packet time reference.

8. The apparatus of Claim 4, wherein said means for outputting further includes means for outputting said data packets in respective logical groups that represent time-synchronous packets from said non-synchronous compute nodes after said grouping criteria has been met.

9. A recordable type medium having a computer program product for collating data in a distributed computer network having non-synchronous compute nodes, said recordable type medium comprising:

computer program code for receiving a plurality of sets of data packets from a plurality of physically separated non-synchronous compute nodes, wherein individual ones of said sets of data packets are provided by individual ones of said non-synchronous compute nodes and wherein individual ones of the plurality of non-synchronous compute nodes comprise individual sources of data packets;

computer program code for inserting said data packets into a software container according to user predetermined rules for determining a logical order for said data packets;

computer program code for locating common groups of said data packets within said software container according to said user predetermined rules;

computer program code for protecting said software container against incomplete groups of said data packets due to system anomalies or quality of service within said distributed computer network according to a grouping criteria; and

computer program code for outputting said data packets in respective logical groups that represent an aggregate packet from at least two of the non-synchronous compute nodes after said grouping criteria has been met.

10. The recordable type medium of Claim 9, wherein said computer program code for inserting further includes computer program code for inserting said data packets into a software container according to individual packet time reference.

11. The recordable type medium of Claim 8, wherein said computer program code for locating further includes computer program code for locating common groups of said data packets within said container according to individual packet time reference.

12. The recordable type medium of Claim 8, wherein said computer program code for outputting further includes computer program code for outputting said data packets in respective logical groups that represent time-synchronous packets from said non-synchronous compute nodes after said grouping criteria has been met.

(9) Evidence Appendix

None.

(10) Related Proceeding Appendix

None.